Computer Science 420

Introduction to Compiler Design

Course Outline

Lecture: Alden 101, MWF 1:30pm to 2:20pm
Lab: Alden 101, Monday 2:30pm to 4:20pm

Web page: [http://cs.allegheny.edu/~jjumadinova/teaching/420.html](http://cs.allegheny.edu/~jjumadinova/teaching/420.html)

Instructor:

Janyl Jumadinova
Office: Alden Room 105
Office phone: 814-332-2881
Email: jjumadinova@allegheny.edu

Office hours:
Monday: 10am-12pm and 4:30pm-5:30pm
Wednesday: 10am-12pm and 2:30pm-3:30pm
Thursday: 4:30pm-5:30pm
Friday: 11am-12pm
and by appointment

To schedule a meeting with me during my office hours, please go to [http://cs.allegheny.edu/~jjumadinova/schedule.html](http://cs.allegheny.edu/~jjumadinova/schedule.html) and click on “Schedule an Appointment...” and select the desirable date and time. You can schedule an appointment outside of my office hours by email.

Textbook:

Also some of the content for this class will also come directly from me.
**The course:**

An introduction to the basic concepts of compiler design and implementation from lexical, syntactic, and semantic analysis to target code generation. Topics are first presented from a high level perspective and then the implementation details are discussed. The semester project includes construction of a compiler for a small language. One laboratory per week.

*Prerequisites: CMPSC 220 and CMPSC 230*

The course page on Sakai will be used for announcements, and for reporting student grades. I will use bitbucket repository for delivering material to you and for getting assignments from you.

**Course Goals:**

- Understand all of the phases of the compiler design including lexical and syntax analysis, type checking, intermediate code generation, register allocation, machine code generation, and assembly and linking.
- Gain practice with using various tools and hardware involved in compiler design.
- Learn how many popular compilers, that are used in practice, work.
- Be able to build your own compiler using the knowledge from previous courses and from the material presented in this course.

**Grading:**

The grade in this class is based on the following categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Participation</td>
<td>5%</td>
</tr>
<tr>
<td>Assignments (homework and labs)</td>
<td>40%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>15%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20%</td>
</tr>
<tr>
<td>Final Project</td>
<td>20%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>100%</td>
</tr>
</tbody>
</table>

These grading categories are defined as follows:

- **Class Participation:** All students are required to actively participate during all of the class sessions. Your participation will involve answering questions posed by me in class, participating in class exercises and discussions, and asking constructive questions of the other members of the class during discussions and presentations.
• **Laboratory Assignments:** Lab assignments invite students to explore different tools and techniques for designing, implementing, evaluating, and documenting various phases of compiler design. To best ensure that students are ready to develop software after graduation, students will complete most of the lab assignments in teams. Many of the lab assignments in this course will expect students to give both a presentation and a demonstration of the software that they created. You must follow proper procedures for submitting your completed lab assignments in order for them to be graded. You will be given instructions on how to do that with your first programming lab assignment.

• **Homework Assignments:** These assignments will give students an opportunity to enhance their understanding of the theoretical material covered during class meetings. These written homework assignments must be done individually; no collaboration or sharing of answers is allowed. Your discussions with others about homework problems must be at a “high” level, dealing with general issues rather than details specific to a particular answer.

• **Midterm Examination:** The midterm exam will cover all of the material in its associated module. The date for the midterm exam will be announced at least one week in advance of the scheduled date, tentatively the midterm will be held the week before the fall break. Unless prior arrangements are made with me, all students will be expected to take the midterm exam on the scheduled date.

• **Final Examination:** The final examination is a cumulative test. The final exam is scheduled for **Friday, December 12 at 7pm**. Unless there are extenuating circumstances, students must take the final examination at the time indicated above.

• **Final Project:** The final project in this class will require you to apply all of the knowledge and skills that you have accumulated during the course of your study to implement a complete compiler for a simple language. The specific details of this task will be given around the middle of the semester. The final project will be completed in groups.

**Assignment Submission:**

All assignments will have a stated due date and are to be turned in electronically on that due date; all assignments must have headers with your name, date and the Honor Code pledge of the student(s) completing the work. You must follow proper procedures for submitting your completed programs in order for them to be graded. You will be given instructions on how to do
that with your first programming assignment.

Late assignments will be accepted for up to one week past the assigned due date with a 15% penalty. All of the late assignments must be submitted at the beginning of the session that is scheduled one week after the due date. Unless special arrangements are made with the course instructor, no assignments will be accepted after the late deadline. For any assignment completed in a group, students must also turn in a one-page document that describes each group member’s contribution to the submitted deliverables.

**Attendance:**

It is mandatory for all students in this course to attend all class and laboratory sessions. The student must notify me of a legitimate absence ahead of the time. Frequent or prolonged absences due to illness should be documented by the student’s doctor, the Health Center, the Dean of Students’ Office, or the office of Student Disability Services. If you need to miss class due to a religious observance, please speak to me in advance to make arrangements to cover material from that day. Students who miss more than five classes or lab sessions without a legitimate excuse will have their final grade in the course reduced by one letter grade. Students who miss more than ten of classes or lab sessions will automatically fail this course.

**Use of Laboratory Facilities and Hardware Tools:**

Throughout the semester, we will experiment with many different software tools that are used during the compiler design phases. Students are required to complete all assignments and the final project while using the department’s laboratory facilities. The course instructor and the systems administrator do not assist students in configuring their personal computers.

We will also experiment with various hardware in this course. The students will be assigned various hardware components for specific assignment and students will be responsible for the assigned hardware over the duration of the given assignments.

**Special Needs and Disability:**

Students with disabilities who believe they may need accommodations in this class are encouraged to contact Disability Services at (814) 332-2898. Disability Services is part of the Learning Commons and is located in Pelletier Library. Please do this as soon as possible to ensure that approved accommodations are implemented in a timely fashion.
Honor Code:

All students enrolled at Allegheny College are bound by the Honor Code. It is expected that your behavior will reflect that commitment. To this end, we expect that you will adhere to the following Department Policy:

Department of Computer Science Honor Code Policy

It is recognized that an important part of the learning process in any course, and particularly in computer science, derives from thoughtful discussions with teachers, student assistants, and fellow students. Such dialogue is encouraged. However, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others, and the student who produces assignments that are identical to, or merely variations on, someone else’s work. It will therefore be understood that all assignments submitted to faculty of the Department of Computer Science are to be the original work of the student submitting the assignment, and should be signed in accordance with the provisions of the Honor Code. Appropriate action will be taken when assignments give evidence that they were derived from the work of others.

You are encouraged to periodically review the specifics of the Honor Code as stated in the College Catalogue, The Compass, and elsewhere.